adr-spec-v0.1

Open Technical Specification

Agent Decision Record

Open Specification, Version 0.1

Document Type	Open Technical Specification
Version	0.1 Founding Draft
Date	February 2026
Status	Public Review - Seeking Comment
Canonical URL	accountability.ai/adr-spec-v0.1
License	Creative Commons CC-BY 4.0
Jurisdiction	Pax Silica Coalition / EU AI Act / US Federal
ISBN	978-1-7389042-0-4

> “Every AI governance tool gives you a dashboard. None give you proof. The ADR is the missing evidentiary layer.”

The Agent Decision Record is an open specification for creating, storing, and verifying tamper-evident records of AI system decisions. It defines the minimum required fields, cryptographic integrity requirements, chain of custody standards, and evidentiary metadata needed for an AI decision record to function as audit-ready evidence before regulators, courts, and coalition compliance bodies.

This specification forms the foundation of the accountability.ai measurement infrastructure. It is published as an open standard that any organization may implement.

founding@accountability.ai	accountability.ai

Table of Contents

Section	Title
01	Executive Summary
02	Background and Motivation

Section	Title
03	Specification Scope
04	Core Field Definitions
05	Cryptographic Integrity Requirements
06	Chain of Custody
07	Reasoning Capture Standards
08	Human Oversight Record
09	Jurisdiction Mapping
10	Versioning and Governance

---

01 Executive Summary

1.1 The ADR in Plain Language

The Agent Decision Record (ADR) is a structured and cryptographically verifiable record generated at the moment an AI system produces a consequential output. It captures what the system decided, who authorized the decision, under what policy, on what evidence, and with what reasoning. All elements are bound into a tamper-evident chain that any party can independently verify.

The ADR addresses a gap that existing AI governance tools have not closed. There is a difference between a system that *appears* accountable and one that can *prove* accountability to a regulator, a court, or a coalition partner. Appearances are not evidence. The ADR is.

This specification defines the minimum fields, cryptographic requirements, and evidentiary standards that an ADR implementation must satisfy. It is intentionally scoped to what is necessary, not what is convenient. The purpose is to produce records that withstand adversarial scrutiny, not records that are merely easy to generate.

1.2 Core Principle

> **Core Principle**
>
> **The ADR is not a logging format. It is an evidentiary artifact.**
>
> The distinction matters. Logs capture what happened for operational purposes. An evidentiary artifact is constructed to survive challenge. Every design decision in this specification reflects that requirement.

---

02 Background and Motivation

2.1 The Evidentiary Gap

AI systems now make consequential decisions at scale: credit approvals, medical triage, export control screening, hiring shortlists, bail recommendations, and content moderation.

In each domain, decisions may be challenged. Challenges come from regulators investigating disparate impact, from plaintiffs alleging discrimination, from coalition partners requiring compliance verification, and from boards demanding accountability after adverse outcomes.

When a challenge arrives, the question is consistent: *Can you prove what your system decided, why it decided it, and that the record has not been altered since?* At most organizations today, the answer is no.

This is not necessarily due to wrongdoing. It is because the infrastructure required to produce that proof does not exist. Logs exist. Dashboards exist. Audit reports exist. None of these constitute evidentiary proof.

2.2 Regulatory Landscape

Regulation	Obligation	ADR Relevance
EU AI Act Article 12	High-risk AI systems must maintain automatic logs enabling post-market monitoring and traceability.	The ADR provides the logging artifact required by Article 12.
EU AI Act Article 13	Transparency requirements require AI outputs to be interpretable by deployers and affected persons.	The reasoning field directly supports the interpretability obligation.
Pax Silica Coalition	Trust but verify. Coalition members must demonstrate AI system integrity to partners.	The ADR hash chain enables independent verification without sharing underlying data.
US SEC / FINRA Rule 17a-4	Broker-dealers must preserve records in non-rewriteable format for examination.	The ADR hash chain architecture supports WORM-style record requirements.
Colorado AI Act (2026)	High-risk AI deployers must conduct impact assessments and maintain decision records.	The ADR provides the required decision record infrastructure.
HIPAA / FDA AI Guidance	AI-assisted clinical decisions must be auditable and traceable to their evidence basis.	The `evidence_basis` and `model_version` fields support clinical traceability.

03 Specification Scope

This specification defines the minimum required structure for a compliant ADR. It does not prescribe implementation technology, storage architecture, or programming language. Any system that produces records conforming to the field definitions, cryptographic requirements, and evidentiary standards defined herein is a conformant ADR implementation.

- What fields an ADR must contain (Core Field Definitions, Section 04)
- How cryptographic integrity must be established and verified (Section 05)
- How chain of custody must be maintained across sequential decisions (Section 06)
- What constitutes compliant reasoning capture (Section 07)
- How human oversight must be recorded when it occurs (Section 08)
- How the specification maps to specific regulatory obligations (Section 09)

This specification intentionally does **not** define how AI systems make decisions, what decisions are permissible, or what outcomes are acceptable. Those questions are for regulators, ethics bodies, and the organizations deploying AI systems. The ADR records what happened. It does not judge it.

---

04 Core Field Definitions

The following fields are required in every conformant ADR. Fields marked **REQUIRED** must be present and non-null. Fields marked **CONDITIONAL** are required when the specified condition is met.

4.1 Identity Fields

Field	Type	Req.	Description
`adr_id`	string	REQUIRED	Globally unique identifier. Time-ordered (ULID recommended). Format: `adr_{26-char-ulid}`. Immutable after generation.
`timestamp`	ISO 8601	REQUIRED	UTC timestamp of decision generation at inference time. Must reflect when the decision was produced, not when the record was written.
`agent_id`	string	REQUIRED	Stable identifier of the AI system or model instance. Must correspond to a registered system in the agent registry.
`agent_version`	string	REQUIRED	Exact version string of the deployed model. Must be sufficient to reproduce the model state at decision time.

4.2 Decision Fields

Field	Type	Req.	Description
`decision_type`	string	REQUIRED	Classification of the decision domain. Examples: `credit_approval`, `clinical_triage`, `export_screen`, `hiring_screen`.
`input_summary`	object	REQUIRED	Pseudonymized summary of inputs. Must not contain raw PII. Must contain sufficient context to reconstruct the decision.
`output`	object	REQUIRED	Exact output produced by the AI system, verbatim. No post-processing or summarization. Includes decision, confidence, and structured metadata.
`confidence`	number	REQUIRED	Confidence score as a percentage (0–100). Must reflect the system's internal certainty metric, not a post-hoc assignment.

4.3 Reasoning Fields

The reasoning field is the most legally significant field in the ADR. It must be captured at inference time. Post-hoc reconstruction of reasoning is **not compliant** with this specification and is insufficient as evidentiary documentation.

Field	Type	Req.	Description
`reasoning`	string	REQUIRED	Human-readable explanation of why the AI system produced this output. Minimum 20 characters. Must be captured at inference, not reconstructed after the fact.
`reasoning_method`	string	CONDITIONAL	Method used to generate the reasoning explanation. Values: `chain_of_thought`, `shap`, `lime`, `rule_trace`, `attention`. Required when confidence < 80%.
`feature_attribution`	object	CONDITIONAL	Feature-level weights contributing to the decision. Required for high-risk AI systems. Format: `{feature_name: weight}`.

4.4 Policy and Authority Fields

Field	Type	Req.	Description
`policy_version`	string	REQUIRED	Version identifier of the governing policy at inference time. Immutable. Policy changes require a new version string.
`jurisdiction`	string[]	REQUIRED	Array of jurisdiction identifiers. Examples: `US-ECOA`, `EU-AIAct-HighRisk`, `Pax-Silica-v1`.
`human_review_required`	boolean	REQUIRED	Whether this decision requires human review before action. Determined by policy at inference time, not assigned after the decision.
`risk_classification`	string	REQUIRED	Risk tier of this decision. Values: `unacceptable`, `high`, `limited`, `minimal`.
`affected_party_id`	string	CONDITIONAL	Pseudonymized identifier of the affected party. Required for decisions affecting identifiable individuals. Never store raw PII.

4.5 Extension Fields

Optional but strongly recommended for high-risk AI systems.

Field	Type	Req.	Description
`data_quality_flags`	object	OPTIONAL	Flags indicating data quality issues at decision time. Example: `{source_stale: true, freshness_hours: 31, threshold_hours: 24}`.
`agent_chain`	string[]	OPTIONAL	Array of `adr_id`s from upstream AI agents. Required for multi-agent pipeline traceability.
`evidence_basis`	object	OPTIONAL	Documentation of evidence or knowledge sources used. For RAG systems: document IDs, retrieval timestamps, relevance scores.
`coalition_compliance`	object	OPTIONAL	Pax Silica coalition compliance metadata. Enables cross-border verification without sharing underlying decision data.

Field	Type	Req.	Description
`remediation_flag`	boolean	OPTIONAL	Indicates this decision has been flagged for remediation review. Once set to `true`, immutable.

---

05 Cryptographic Integrity Requirements

Every ADR must include three cryptographic elements: a record hash, a previous hash linking it to the prior record in the chain, and a signature binding it to the generating infrastructure. Together these make the record tamper-evident and independently verifiable.

Field	Type	Req.	Description
`record_hash`	string	REQUIRED	SHA-256 hash of the canonical serialization of this ADR. Exact field ordering specified to ensure deterministic reproduction.
`previous_hash`	string	REQUIRED	The `record_hash` of the preceding ADR in this `agent_id` chain. Genesis record uses `SHA-256("ADR-GENESIS-{agent_id}-{date}")`.
`signature`	string	REQUIRED	Ed25519 digital signature of the `record_hash`, produced by the infrastructure key registered for this agent. Enables third-party verification.
`chain_sequence`	integer	REQUIRED	Monotonically increasing integer per `agent_id`. Gaps in sequence indicate deleted records. Combined with `previous_hash` for dual tamper evidence.

5.1 Hash Chain Integrity

$$H(n) = \text{SHA256}(\text{Payload}(n) \parallel H(n-1))$$

Where $H(n)$ is the record hash, $\text{Payload}(n)$ is the canonical serialization, and $\parallel$ denotes concatenation.

5.2 Canonical Serialization

The canonical form serializes fields in the order defined in Section 04, with no whitespace, using UTF-8 encoding. Fields included in the canonical hash are: `adr_id`, `timestamp`, `agent_id`, `agent_version`, `decision_type`, `output`, `confidence`, `reasoning`, `policy_version`, `jurisdiction`, `previous_hash`, `chain_sequence`. Extension fields are excluded to preserve forward compatibility.

5.3 Key Management

Production implementations **must** use Ed25519 key pairs managed by a hardware security module (HSM) or equivalent. The public key must be published at a well-known endpoint to enable third-party verification. Key rotation must produce a new genesis record in the chain.

---

06 Chain of Custody

The ADR chain of custody is established through the combination of sequential `chain_sequence` integers and cryptographic hash linkage. Every record references the hash of its predecessor, making any alteration or deletion detectable.

6.1 Chain Structure

$\ADRn-1 \rightarrow ADRn \rightarrow ADRn+1$ $seq: n-1 seq: n seq: n+1$
$prev_hash \rightarrow prev_hash \rightarrow \ldots \$

6.2 Tamper Detection

- Deletion of a record: creates a gap in `chain_sequence`. Detectable by any verifier.
- **Alteration of a record:** changes `record_hash`, breaking the hash link to the successor. Detectable by any verifier.
- **Insertion of a fabricated record:** requires re-signing with the infrastructure key. Detectable if the public key is registered.
- **Alteration of reasoning after the fact:** changes `record_hash`. This is the primary protection against post-hoc rationalization.

6.3 Immutability Requirements

Conformant ADR storage must enforce append-only behavior. No `UPDATE` or `DELETE` operations are permitted on ledger records after insertion. This must be enforced at the storage layer, not the application layer. Row-level security policies, WORM storage, or equivalent controls satisfy this requirement.

The immutability requirement is not a technical preference. It is the foundation of the evidentiary standard. A record that can be altered is not evidence. The absence of update and delete permissions must be verifiable by auditors.

---

07 Reasoning Capture Standards

The reasoning field is the most frequently challenged field in an adversarial examination. The following standards govern what constitutes compliant reasoning capture.

7.1 Contemporaneous Requirement

Reasoning must be captured at the moment of inference, as part of the same transaction that produces the decision output. Reasoning generated after the fact—even if accurate—does not satisfy this requirement. The timestamp of the reasoning must match the timestamp of the decision.

7.2 Minimum Content Standard

A compliant reasoning field **must**:

- State what the AI system decided and at what confidence level.
- Identify the primary factors that drove the decision, with direction (supporting or opposing).
- Flag any data quality issues present at decision time.
- Identify whether a near-miss condition existed (decision within threshold of an alternative outcome).
- State the policy basis for the decision and any human review requirement.

A reasoning field that states only "approved" or "denied" is not compliant. A reasoning field that reconstructs a plausible explanation after the fact is not compliant and constitutes a falsification of the evidentiary record.

---

08 Human Oversight Record

When a human reviewer intervenes in an AI decision, that intervention must be recorded as a separate artifact linked to the originating ADR. The human oversight record preserves the original AI output even when the human modifies or overrides it.

Field	Type	Req.	Description
`ledger_event_id`	uuid	REQUIRED	Reference to the ADR that triggered this human review.
`reviewer_id`	string	REQUIRED	Pseudonymized identifier of the human reviewer. Must be traceable to a named individual in a separate, access-controlled registry.
`outcome`	string	REQUIRED	Result of human review. Values: `approved`, `denied`, `modified`, `escalated`.
`original_output`	object	REQUIRED	The AI system's original output before human modification. Must be preserved verbatim even when overridden.
`modified_output`	object	CONDITIONAL	The human-modified output. Required when outcome is

Field	Type	Req.	Description
			`modified`. Records the human's intervention verbatim.
`human_override_delta`	object	CONDITIONAL	Structured diff between `original_output` and `modified_output`. Captures direction and magnitude of human correction. Required when outcome is `modified`.
`reviewer_notes`	string	OPTIONAL	Free-text notes documenting the basis for the reviewer's decision. Strongly recommended for `modified` and `escalated` outcomes.
`reviewed_at`	ISO 8601	REQUIRED	UTC timestamp of human review completion. This is the legal timestamp of the human intervention.
`review_duration_secs`	integer	OPTIONAL	Time elapsed between assignment and completion of review. Used to detect rubber-stamping patterns in compliance audits.

---

09 Jurisdiction Mapping

The ADR specification is jurisdiction-agnostic at the core field level. The `jurisdiction` field and `coalition_compliance` extension enable jurisdiction-specific metadata without breaking the core standard.

Regulatory Obligation	ADR Field(s)	Notes
EU AI Act Art. 12 - Logging	`adr_id`, `timestamp`, `agent_id`, `decision_type`, `output`	All high-risk systems
EU AI Act Art. 13 - Transparency	`reasoning`, `feature_attribution`, `confidence`	Interpretability requirement
Reg BI - Best Interest	`reasoning`, `policy_version`, `human_review_required`	Wealth management suitability
FINRA Rule 2111 - Suitability	`reasoning`, `data_quality_flags`, `confidence`	Risk assessment freshness
ECOA / Fair Housing	`feature_attribution`, `reasoning`, `affected_party_id`	Disparate impact analysis
Pax Silica Verification	`record_hash`, `signature`, `chain_sequence`	Cross-border verification

Regulatory Obligation	ADR Field(s)	Notes
HIPAA / FDA AI	`evidence_basis`, `agent_version`, `reasoning_method`	Clinical traceability
Colorado AI Act (2026)	All core fields + `risk_classification`	High-risk deployers
SEC Rule 17a-4 / WORM	Append-only storage + `record_hash` chain	Storage architecture

---

10 Versioning and Governance

The ADR specification uses **semantic versioning**.

- **Major** version increments indicate breaking changes to field structure or cryptographic requirements.
- **Minor** version increments add new fields or clarify existing ones in a backward-compatible manner.
- **Patch** version increments are editorial corrections.

This document is version **0.1**, a founding draft released for public comment. Transition to version 1.0 will occur only after:

- A public comment period of no less than 90 days.
- Review by at least three independent regulatory experts.
- Pilot implementation by at least three organizations across different sectors.

10.1 Governance Structure

Governance of the specification will transition to a **multi-stakeholder body** prior to version 1.0. The structure will ensure that no single organization, including `accountability.ai`, controls the standard. The standard must be owned by the ecosystem it serves.

10.2 How to Engage

This specification is published for public comment. Input is sought from:

- Regulatory practitioners and former examiners who can stress-test the evidentiary standards.
- Technical implementers who can identify gaps in the cryptographic and systems design.
- Organizations planning to implement the ADR who can identify practical barriers.
- Legal practitioners who can evaluate the admissibility framework.

General enquiries	founding@accountability.ai
Technical comment	spec@accountability.ai

> "The standard that defines AI accountability must itself be accountable. Submit your challenge. We will answer it on the record."

Agent Decision Record – Open Specification v0.1

Published	**February 2026**
Canonical URL	accountability.ai/adr-spec-v0.1
ISBN	978-1-7389042-0-4

Copyright © 2026 Accountability.ai
This work is licensed under **Creative Commons Attribution 4.0 International (CC-BY 4.0)**.

You are free to share and adapt this specification for any purpose, including commercial use, provided appropriate credit is given and a link to the license is included.

Attribution: *"Agent Decision Record Specification v0.1, Accountability.ai, February 2026, accountability.ai/adr-spec-v0.1"*

The `accountability.ai` name and infrastructure are the reference implementation. The standard belongs to the ecosystem it serves.

Pax Silica Coalition · EU AI Act · US Federal
Status: Public Review, Seeking Comment